```
                SUBTTL  DEFINITIONS AND EQUATES
                TITLE   ANS8811.MAC  21.11.88  LAST EDIT 13.12.88
;LAST EDIT as above in title line
;
;
;       !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
;       ! INCOMPLETELY TESTED - FOR GUIDANCE ONLY !
;       !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
;
;
;        (c) 1988 DM PARKINS, GREENBANK ELECTRONICS
;                        051-645 3391
;
;This is a collection of demonstration routines and
;initialisation tables for use with the Interak "QS-1"
;Quad Serial RS-232 Card
;
;Four front panel RS-232 ports are provided, derived from
;the internal (chip) SIO/DART channels as follows
;(view from outside of computer, front panel vertical)
;
;       RS-232 Port 0  (from SIO0 Channel A)  Top LHS
;       RS-232 Port 1  (from SIO0 Channel B)  Top RHS
;       RS-232 Port 2  (from SIO1 Channel A)  Bottom LHS
;       RS-232 Port 3  (from SIO1 Channel B)  Bottom RHS
;
;Typical output connector links for use of the QS-1 board
;and these routines with a printer would be as follows
;
;All links generally in the "DCE" (left hand) positions,
;except for the top groups of three, where one should be
;fitted to link 1-2 in the lowest of the 3 possible positions
;in each of the four groups of 3.
;
;For handshaking instructions read the next paragraph.
;
;if RTS and CTS handshaking is not used fit a vertical
;link to "U-turn" these signals
;
;if DTR and DSR handshaking is not used fit a vertical
;link to "U-turn" these signals
;
;During initial testing of these routines, our advice is
;to fit the vertical "U-turn" links in each group, at which
;stage a so called "3-wire" RS-232 link can be made to
;the peripheral device - no hardware handshaking then
```

```
;being operative. Work out which hardware handshaking
;method (if any) the peripheral requires, by trial
;and error - ;fit horizontal links for RTS, CTS, DTR, DSR
;until communication is achieved, without lost characters.
;
;XON/XOFF, EXT/ACK and any other software handshaking
;is not demonstrated in these routines, which are merely
;intended to show how the QS-1 is operated at its most
;basic level. More information on software handshaking
;techniques, and a demonstration print driver routine for
;the QS-1 card is contained in the document SN-U8811,
;written by user JW Bowes - "Printer Driver for QS-1"

;Although some hardware exists on the QS-1 card to allow
;experiments with control via interrupts, the board is only
;sold as being suitable for use in a polled or non-interrupt
;environment.  This makes the writing and debugging of
;software considerably easier, and perhaps surprisingly does
;not slow down the total system operation.  The secret of
;working by polling the various channels is to ensure that
;(unless you want to) that you never sit in a loop waiting
;for something to happen. Use the Input Status and Output
;Status subroutines provided here for your software to
;dash in and out to see if attention is needed; and if not
;do something useful instead of waiting. (The FIFO buffers
;built into the SIO/DART chips will store the data safely
;for a while if your software is away doing other things.)
;
;
;This listing is divided up into the following sections
;
;       1       Introduction (which you are reading)
;       2       Port Definitions, equates etc
;       3       SIO/DART Initialisation Tables
;       4       Subroutines
;               i       Initialisation (INITQS)
;               ii      Test Input status (INSA0,B0,A1,B1)
;               iii     Input a byte (INCA0,B0,A1,B1)
;               iv      Test Output status (OUTSA0,B0,A1,B1)
;               v       Output a byte (OUTCA0,B0,A1,B1)
;       5       Short Demonstration program (DEMO)
;
;-------------------------------------------------------------

;               Start of Listing
;
```

```
                        ;(The following directive suits the Microsoft Macro 80
                        ;assembler program used, delete or change it to suit your
                        ;own assembler).  Labels are marked by two colons to make
                        ;them "global" for the Macro 80 assembler, as an easy way
                        ;to force the generation of a complete symbol table.  You
                        ;should use whatever suits your assembler)

                        .Z80                        ;Z80 mnemonics

                        ORG     0000H               ;alter origin to suit the
                                                    ;subroutine and table areas
                                                    ;in your own program

0000' C3 0111'                  JP      DEMO        ;(DEMO is a demonstration of
                                                    ;of the subroutines in action)

                        ;-------------------------------------------------------------
                        ;PORT ALLOCATIONS FOR BAUD RATE AND SIO CHIP(S)

                        ;QSPORT EQU     ppppp000B   ;choose eight consecutive
                                                    ;ports, (starting on a logical
                                                    ;boundary),
                                                    ;ie ppppp000 to ppppp111

0000                    QSPORT  EQU     0           ;EXAMPLE Port 0

                        ;CARDNO EQU     ccccc000B   ;QS-1 Card number "ccccc000"
                                                    ;nnnnn = 00000 to 11111
                                                    ;(choose one of the 32 possible
                                                    ;cards which can use the
                                                    ;previously defined
                                                    ;ppppp000-ppppp111 ports)
                                                    ;In practice if you only have
                                                    ;one QS-1 card in the system
                                                    ;it can be set to be card
                                                    ;number 00H - in which case
                                                    ;you will need
                                                    ;no initialisation,
                                                    ;because the 74HC175s on the
                                                    ;card power on to all zeros
                                                    ;anyway

0000                    CARDNO  EQU     0           ;EXAMPLE Card number 0

0000                    SIO0    EQU     CARDNO+0    ;first SIO chip
0001                    SIO1    EQU     CARDNO+1    ;second SIO chip
```

```
0004                    BAUD0   EQU     CARDNO+4        ;first baud rate chip
0005                    BAUD1   EQU     CARDNO+5        ;second baud rate chip
0007                    QSCTRL  EQU     QSPORT+7        ;QS-1 card control port


                        ;SIO REGISTER ADDRESSES (identical for both SIO0 and SIO1)

0000                    DATAA   EQU     QSPORT+0        ;Data registers A
0001                    CTRLA   EQU     DATAA +1        ;Control registers A
0002                    DATAB   EQU     QSPORT+2        ;Data registers B
0003                    CTRLB   EQU     DATAB +1        ;Control registers B


                        ;BAUD RATE CHIP ADDRESS (actually occupies 4 addresses 0-3)

0000                    BDRATE  EQU     QSPORT+0        ;High nibble is "B", low "A"


                        ;CONSTANTS
                        ;8116 register values for given baud rates
0000                    BD0050  EQU     0000B           ;  50.0 baud
0001                    BD0075  EQU     0001B           ;  75.0 baud
0002                    BD0110  EQU     0010B           ; 110.0 baud
0003                    BD0134  EQU     0011B           ; 134.5 baud
0004                    BD0150  EQU     0100B           ; 150.0 baud
0005                    BD0300  EQU     0101B           ; 300.0 baud
0006                    BD0600  EQU     0110B           ; 600.0 baud
0007                    BD1200  EQU     0111B           ;1200.0 baud
0008                    BD1800  EQU     1000B           ;1800.0 baud
0009                    BD2000  EQU     1001B           ;2000.0 baud
000A                    BD2400  EQU     1010B           ;2400.0 baud
000B                    BD3600  EQU     1011B           ;3600.0 baud
000C                    BD4800  EQU     1100B           ;4800.0 baud
000D                    BD7200  EQU     1101B           ;7200.0 baud
000E                    BD9600  EQU     1110B           ;9600.0 baud
000F                    BD19K2  EQU     1111B           ;  19.2 kilobaud

                        ;for first chip
                        ;BRA0    EQU     BDaaaa          ;choose from above list
0005                    BRA0    EQU     BD0300          ;EXAMPLE  300 baud

                        ;BRB0    EQU     BDbbbb          ;choose from above list
000E                    BRB0    EQU     BD9600          ;EXAMPLE 9600 baud

00E5                    BRAB0   EQU     BRA0+16*BRB0    ;together
```

```
                        ;for second chip
                        ;BRA1   EQU     BDcccc          ;choose from above list
000E                    BRA1    EQU     BD9600          ;EXAMPLE 9600 baud

                        ;BRB1   EQU     BDdddd          ;choose from above list
0007                    BRB1    EQU     BD1200          ;EXAMPLE 1200 baud

007E                    BRAB1   EQU     BRA1+16*BRB1    ;together


                        ;-----------------------------------------------------------
                        SUBTTL SIO/DART CHIP INITIALISATION TABLES
                        PAGE
```

```
                        ;*****************************************************************************
                        ;SUBTTL SIO/DART CHIP INITIALISATION TABLES
                        ;*****************************************************************************

                        ;These tables contain the set of bytes which would
                        ;be sent to the SIO/DART chips on the QS-1 card during
                        ;initialisation.  The data here is typical only, and of
                        ;course may be varied to suit the user's specific needs.
                        ;Consult the Z80 SIO or DART data sheets first though, for
                        ;the details of what the commands do, and in what
                        ;sequence(s) they may be sent. (Do not vary the order of
                        ;the commands in the tables unless you know what you are
                        ;doing, as sometimes the order of events is critical.)

                        ;-------------------------------------------------------------
                        ;note - during this initialisation, outputs RTS and DTR
                        ;are both set active, to
                        ;
                        ;(i) suit those peripherals which require these handshakes
                        ;
                        ;                       or,
                        ;
                        ;(ii) to link back via "U-turn links" so that the following
                        ;QS-1 handshaking inputs are forced active
                        ;
                        ;       force CTS input by "U-turn link" from RTS output
                        ;       force DCD input by "U-turn link" from DTR output
                        ;
                        ;         (Vertical links on QS DCE/DTE jumper arrays)
                        ;
                        ;-------------------------------------------------------------

                        ;change the ORG statement below to suit a suitable address for
                        ;tables in your program

                        ORG     $


                        ;Example initialisation table for "A" port of SIO chip "0"
0003'  18               CTS0A:  DEFB    00011000B       ;WR0 Channel Reset
0004'  10                       DEFB    00010000B       ;WR0, Reset External/Status Ints
0005'  04                       DEFB    04H             ;Point to WR4
0006'  4C                       DEFB    01001100B       ;X16 CLK, No Parity,
```

```
                                                        ;2 Stop bits
0007'  03                       DEFB    03H             ;Point to WR3
0008'  C1                       DEFB    11000001B       ;8 Bits/char, receive,
                                                        ;No auto enable
0009'  05                       DEFB    05H             ;Point to WR5
000A'  EA                       DEFB    11101010B       ;Set RTS, DTR,8 bits/char,
                                                        ;enable XMIT
000B'  10                       DEFB    00010000B       ;WR0, Reset External/Status Ints
000C'  01                       DEFB    01H             ;Point to WR1
000D'  00                       DEFB    00000000B       ;Disable external and transmit
                                                        ;ints

                        ;Example initialisation table for "B" port of SIO chip "0"
000E'  18               CTS0B:  DEFB    00011000B       ;WR0 Channel Reset
000F'  10                       DEFB    00010000B       ;WR0, Reset External/Status Ints
0010'  02                       DEFB    02H             ;Point to WR2 (Channel B only)
0011'  00                       DEFB    00H             ;Interrupt Vector (not used!)
0012'  04                       DEFB    04H             ;Point to WR4
0013'  4C                       DEFB    01001100B       ;X16 CLK, No Parity,
                                                        ;2 Stop bits
0014'  03                       DEFB    03H             ;Point to WR3
0015'  C1                       DEFB    11000001B       ;8 Bits/char, receive,
                                                        ;No auto enables
0016'  05                       DEFB    05H             ;Point to WR5
0017'  EA                       DEFB    11101010B       ;Set RTS, DTR,8 bits/char,
                                                        ;enable XMIT
0018'  10                       DEFB    00010000B       ;WR0, Reset External/Status Ints
0019'  01                       DEFB    01H             ;Point to WR1
001A'  00                       DEFB    00000000B       ;Disable external and transmit
                                                        ;ints

                        ;Example initialisation table for "A" port of SIO chip "1"
001B'  18               CTS1A:  DEFB    00011000B       ;WR0 Channel Reset
001C'  10                       DEFB    00010000B       ;WR0, Reset External/Status Ints
001D'  04                       DEFB    04H             ;Point to WR4
001E'  4C                       DEFB    01001100B       ;X16 CLK, No Parity,
                                                        ;2 Stop bits
001F'  03                       DEFB    03H             ;Point to WR3
0020'  C1                       DEFB    11000001B       ;8 Bits/char, receive,
                                                        ;No auto enable
0021'  05                       DEFB    05H             ;Point to WR5
0022'  EA                       DEFB    11101010B       ;Set RTS, DTR,8 bits/char,
                                                        ;enable XMIT
0023'  10                       DEFB    00010000B       ;WR0, Reset External/Status Ints
0024'  01                       DEFB    01H             ;Point to WR1
0025'  00                       DEFB    00000000B       ;Disable external and transmit
```

```
                                                    ;ints

                        ;Example initialisation table for "B" port of SIO chip "1"
0026'  18               CTS1B:  DEFB    00011000B       ;WR0 Channel Reset
0027'  10                       DEFB    00010000B       ;WR0, Reset External/Status Ints
0028'  02                       DEFB    02H             ;Point to WR2 (Channel B only)
0029'  00                       DEFB    00H             ;Interrupt Vector (not used!)
002A'  04                       DEFB    04H             ;Point to WR4
002B'  4C                       DEFB    01001100B       ;X16 CLK, No Parity,
                                                        ;2 Stop bits
002C'  03                       DEFB    03H             ;Point to WR3
002D'  C1                       DEFB    11000001B       ;8 Bits/char, receive,
                                                        ;No auto enable
002E'  05                       DEFB    05H             ;Point to WR5
002F'  EA                       DEFB    11101010B       ;Set RTS, DTR,8 bits/char,
                                                        ;enable XMIT
0030'  10                       DEFB    00010000B       ;WR0, Reset External/Status Ints
0031'  01                       DEFB    01H             ;Point to WR1
0032'  00                       DEFB    00000000B       ;Disable external and transmit
                                                        ;ints

0033'                   CTSEND: ;end of example control tables for SIO


                        ;-----------------------------------------------------------
                        SUBTTL SUBROUTINES
                        PAGE
```

```
                        ;*****************************************************************************
                        ;SUBTTL SUBROUTINES
                        ;*****************************************************************************


                        ;change the ORG statement below to suit a suitable address
                        ;for subroutines in your program

                        ORG     $

0033'                   INITQS::;initialisation of QS-1 card - shown here as a
                               ;subroutine, but often can be built into the
                               ;user's program, as typically it would be called
                               ;only once.  Use of a subroutine for initialisation
                               ;is useful if there are several occasions during
                               ;the course of a program where the SIO/DART chips
                               ;need reinitialising to new values.  In such cases
                               ;the initialisation tables had better be in RAM
                               ;so that they can be set up to the required values
                               ;before initialisation

                               ;the values used (baud rate and so on) are fixed
                               ;to defined values here to demonstrate the
                               ;initialisation as "simply" as possible - if the
                               ;user wants to vary any values on the fly, (eg
                               ;baud rates, parity, no of bits, etc) then of course
                               ;he can alter whatever he wants


                               ;Step 1 - initialise Baud rate chips directly
                               ;select first baud rate generator chip, BAUD0
0033'   3E 04                  LD      A,BAUD0
0035'   D3 07                  OUT     (QSCTRL),A      ;Select first Baud rate chip
0037'   3E E5                  LD      A,BRAB0         ;
0039'   D3 00                  OUT     (BDRATE),A      ;load its registers
                               ;next baud rate generator chip, BAUD1
003B'   3E 05                  LD      A,BAUD1
003D'   D3 07                  OUT     (QSCTRL),A      ;Select second Baud rate chip
003F'   3E 7E                  LD      A,BRAB1
0041'   D3 00                  OUT     (BDRATE),A      ;load its register

                               ;Step 2 - initialise SIO chips from tables
                               ;select first SIO chip, SIO0
```

```
0043'  3E 00                    LD      A,SIO0
0045'  D3 07                    OUT     (QSCTRL),A      ;Select first SIO

                                ;Channel A first
0047'  06 0B                    LD      B,CTS0B-CTS0A   ;length SIO0 Channel A table
0049'  0E 01                    LD      C,CTRLA         ;SIO0 A Control Port
004B'  21 0003'                 LD      HL,CTS0A        ;Pointer to Control for
                                                        ;Table SIO0 Ch A
004E'  ED B3                    OTIR                    ;SIO0 Ch A initialised

                                ;Channel B next
0050'  06 0D                    LD      B,CTS1A-CTS0B   ;length SIO0 Channel B table
0052'  0E 03                    LD      C,CTRLB         ;SIO B Control Port
0054'  21 000E'                 LD      HL,CTS0B        ;Control Table SIO0 Ch B
0057'  ED B3                    OTIR                    ;SIO0 Ch A initialised

                                ;select second SIO chip, SIO1
0059'  3E 01                    LD      A,SIO1
005B'  D3 07                    OUT     (QSCTRL),A      ;Select second SIO

                                ;Channel A
005D'  06 0B                    LD      B,CTS1B-CTS1A   ;length SIO1 Channel A table
005F'  0E 01                    LD      C,CTRLA         ;SIO1 A Control Port
0061'  21 001B'                 LD      HL,CTS1A        ;Control Table SIO1 Ch A
0064'  ED B3                    OTIR                    ;SIO1 Ch A initialised

                                ;Channel B
0066'  06 0D                    LD      B,CTSEND-CTS1B  ;length SIO1 Channel B table
0068'  0E 03                    LD      C,CTRLB         ;SIO1 B Control Port
006A'  21 0026'                 LD      HL,CTS1B        ;Control Table SIO1 Ch B
006D'  ED B3                    OTIR                    ;SIO1 Ch B initialised
006F'  C9                       RET     ;end of initialisation subroutine

                        ;-----------------------------------------------------------
                        ;subroutines for determining input status of
                        ;the chosen channel of the chosen SIO.  Exit with A=NZ if
                        ;there is data available, Z otherwise
                        ;four entry points, one exit

0070'                   INSA0:: ;(find input status of SIO0 channel A)
0070'  3E 00                    LD      A,SIO0          ;select SIO0
0072'  0E 01                    LD      C,CTRLA         ;control register "A"
0074'  C3 008C'                 JP      INSTAT
0077'                   INSB0:: ;(find input status of SIO0 channel B)
0077'  3E 00                    LD      A,SIO0          ;select SIO0
0079'  0E 03                    LD      C,CTRLB         ;control register "B"
```

```
007B'  C3 008C'                    JP      INSTAT
007E'                        INSA1:: ;(find input status of SIO1 channel A)
007E'  3E 01                       LD      A,SIO1          ;select SIO1
0080'  0E 01                       LD      C,CTRLA         ;control register "A"
0082'  C3 008C'                    JP      INSTAT
0085'                        INSB1:: ;(find input status of SIO1 channel B)
0085'  3E 01                       LD      A,SIO1          ;select SIO1
0087'  0E 03                       LD      C,CTRLB         ;control register "B"
0089'  C3 008C'                    JP      INSTAT

008C'                        INSTAT::;subroutine to find input status of given channel
                                    ;in (C) of given SIO in (A)
008C'  D3 07                       OUT     (QSCTRL),A      ;Select chosen SIO
008E'  3E 10                       LD      A,010H          ;reset external/status (makes
                                                           ;DCD etc bits valid in SIO)
0090'  ED 79                       OUT     (C),A           ;issue command to control reg
0092'  ED 78                       IN      A,(C)           ;input status byte
0094'  E6 01                       AND     01H             ;keep only Bit 1 (DAV)
                                    ;on exit A is non zero (and Z flag is 0) if
                                    ;SIO input status shows data available
                                    ;(Z otherwise)
0096'  C9                          RET     ;end of input status subroutines

                             ;-------------------------------------------------------------

                             ;subroutines for inputting a byte from
                             ;the chosen SIO, chosen channel.
                             ;four entry points, one exit


                             ;These loops forever if
                             ;there is no data, so you should call the status
                             ;check (INSA0, A1, B0, B1) before calling one of these

                             ;a possible refinement would be to include a call to a
                             ;time out subroutine, so that there would be a way to get
                             ;out of this infinite loop if the routine was ever
                             ;inadvertently entered to output to a device which
                             ;happened never to have data available

0097'                        INCA0:: ;(input a byte from SIO0 channel A)
0097'  3E 00                       LD      A,SIO0          ;select SIO0
0099'  0E 01                       LD      C,CTRLA         ;control register "A"
009B'  C3 00B3'                    JP      INCH

009E'                        INCB0:: ;(input a byte from SIO0 channel B)
```

```
009E'  3E 00                    LD      A,SIO0          ;select SIO0
00A0'  0E 03                    LD      C,CTRLB         ;control register "B"
00A2'  C3 00B3'                 JP      INCH

00A5'                   INCA1:: ;(input a byte from SIO1 channel A)
00A5'  3E 01                    LD      A,SIO1          ;select SIO1
00A7'  0E 01                    LD      C,CTRLA         ;control register "A"
00A9'  C3 00B3'                 JP      INCH

00AC'                   INCB1:: ;(input a byte from SIO1 channel B)
00AC'  3E 01                    LD      A,SIO1          ;select SIO1
00AE'  0E 03                    LD      C,CTRLB         ;control register "B"
00B0'  C3 00B3'                 JP      INCH

00B3'                   INCH::  ;subroutine
00B3'  CD 008C'                 CALL    INSTAT          ;get input status
00B6'  CA 00B3'                 JP      Z,INCH          ;wait until ready
00B9'  0D                       DEC     C               ;DATA port is (CTRL port -1)
00BA'  ED 78                    IN      A,(C)           ;get data from port
00BC'  C9                       RET     ;end of character input subroutines

                        ;-------------------------------------------------------------
                        ;subroutines for determining output status of
                        ;the chosen channel of the chosen SIO.  Exit with A=Z if the
                        ;channel is not busy, NZ otherwise
                        ;four entry points, one exit

00BD'                   OUTSA0:: ;(find output status of SIO0 channel A)
00BD'  3E 00                    LD      A,SIO0          ;select SIO0
00BF'  0E 01                    LD      C,CTRLA         ;control register "A"
00C1'  C3 00D9'                 JP      OSTAT
00C4'                   OUTSB0:: ;(find output status of SIO0 channel B)
00C4'  3E 00                    LD      A,SIO0          ;select SIO0
00C6'  0E 03                    LD      C,CTRLB         ;control register "B"
00C8'  C3 00D9'                 JP      OSTAT
00CB'                   OUTSA1:: ;(find output status of SIO1 channel A)
00CB'  3E 01                    LD      A,SIO1          ;select SIO1
00CD'  0E 01                    LD      C,CTRLA         ;control register "A"
00CF'  C3 00D9'                 JP      OSTAT
00D2'                   OUTSB1:: ;(find output status of SIO1 channel B)
00D2'  3E 01                    LD      A,SIO1          ;select SIO1
00D4'  0E 03                    LD      C,CTRLB         ;control register "B"
00D6'  C3 00D9'                 JP      OSTAT

00D9'                   OSTAT:: ;subroutine to find output status of given channel
                                ;in (C) of given SIO in (A)
```

```
00D9'  D3 07                    OUT     (QSCTRL),A      ;Select chosen SIO
00DB'  3E 10                    LD      A,010H          ;reset external/status (makes
                                                        ;DCD etc bits valid in SIO)
00DD'  ED 79                    OUT     (C),A           ;issue command to control reg
00DF'  ED 78                    IN      A,(C)           ;input status byte
00E1'  E6 2C                    AND     00101100B       ;keep only Bit 5 (CTS),
                                                        ;          Bit 3 (DCD),
                                                        ;          Bit 2 (TBMT)
00E3'  EE 2C                    XOR     00101100B       ;test all bits for 1's
                                ;on exit A is non zero (and Z flag is reset) if
                                ;SIO output status is OK for sending, ie not busy
                                ;(Z otherwise)
00E5'  C9                       RET     ;end of output status subroutines

                        ;-------------------------------------------------------------
                        ;subroutines for outputting a byte to
                        ;the chosen SIO, chosen channel.
                        ;four entry points, one exit

                        ;These loop forever if
                        ;the channel is not ready, so you should call the status
                        ;check (OUTSA0, A1, B0, B1) before calling these routines

                        ;a possible refinement would be to include a call to a
                        ;time out subroutine, so that there would be a way to get
                        ;out of this infinite loop if the routine was ever
                        ;inadvertently entered to output to a device which
                        ;happened to be permanently busy

00E6'                   OUTCA0:: ;(output a byte to SIO0 channel A)
00E6'  57                       LD      D,A             ;save data
00E7'  3E 00                    LD      A,SIO0          ;select SIO0
00E9'  0E 01                    LD      C,CTRLA         ;control register "A"
00EB'  C3 0106'                 JP      OUTCH

00EE'                   OUTCB0:: ;(output a byte to SIO0 channel B)
00EE'  57                       LD      D,A             ;save data
00EF'  3E 00                    LD      A,SIO0          ;select SIO0
00F1'  0E 03                    LD      C,CTRLB         ;control register "B"
00F3'  C3 0106'                 JP      OUTCH

00F6'                   OUTCA1:: ;(output a byte to SIO1 channel A)
00F6'  57                       LD      D,A             ;save data
00F7'  3E 01                    LD      A,SIO1          ;select SIO1
00F9'  0E 01                    LD      C,CTRLA         ;control register "A"
00FB'  C3 0106'                 JP      OUTCH
```

```
00FE'                   OUTCB1:: ;(output a byte to SIO1 channel B)
00FE'  57                       LD      D,A             ;save data
00FF'  3E 01                    LD      A,SIO1          ;select SIO1
0101'  0E 03                    LD      C,CTRLB         ;control register "B"
0103'  C3 0106'                 JP      OUTCH

0106'                   OUTCH:: ;subroutine
0106'  CD 00D9'                 CALL    OSTAT           ;get output status
0109'  C2 0106'                 JP      NZ,OUTCH        ;wait until ready
010C'  0D                       DEC     C               ;DATA port is (CTRL port -1)
010D'  7A                       LD      A,D             ;restore data
010E'  ED 79                    OUT     (C),A           ;send data
0110'  C9                       RET     ;end of character output subroutines


                        ;*****************************************************************************
                        ;END OF SUBROUTINES
                        ;*****************************************************************************


                        SUBTTL DEMO PROGRAM
                        PAGE
```

```
;*****************************************************************************
;SUBTTL DEMO PROGRAM
;*****************************************************************************

;this demonstration program suits one particular hardware
;set up which was convenient for the author to arrange.
;The four front-panel RS-232 Ports were allocated thus
;
;       RS-232 Port 0  300 baud input from DEC terminal
;       RS-232 Port 1 9600 baud output to a cable
;       RS-232 Port 2 9600 baud input from the cable
;       RS-232 Port 3 1200 baud output to serial printer
;
;The demonstration is that any character typed on the
;terminal passes through all four ports of the QS-1 before
;being printed at the printer.
;
;In this experimental set up the following links were made
;on the RS-232 patch areas on the QS-1
;
;Port #0
;       P3      No Link
;       P4      No Link
;       P5  pin 1 Links P5  pin 2       (Horizontal, DSR)
;       P6  pin 3 Links P7  pin 3       (Vertical, DCD-DTR)
;       P8  pin 3 Links P9  pin 3       (Vertical, CTS-RTS)
;       P10 pin 2 Links P10 pin 3       (Horizontal, RxD)
;       P11 pin 2 Links P11 pin 3       (Horizontal, TxD)
;
;
;Port #1,2,3 as above, using appropriate pin assemblies, thus
;
;       Port #        Pin assemblies
;
;         0       3  4  5  6  7  8  9 10 11
;         1      18 19 20 21 22 23 24 25 26
;         2      37 38 39 40 41 42 43 44 45
;         3      52 53 54 55 56 57 58 59 60
;

;change the ORG statement below to suit a suitable address
;for the demo program, or your own alternative
```

```
                                ORG     $

0111'                           DEMO::

                                ;DEMO program takes data in from RS-232 Port 0 at 300 baud,
                                ;outputs it to RS-232 Port 1 at 9600 baud, inputs it from
                                ;RS-232 Port 2 at 9600 baud, and outputs it from RS-232
                                ;Port 3 at 1200 baud

0111'  31 015F'                         LD      SP,STACK        ;establish a stack
0114'  CD 0033'                         CALL    INITQS          ;initialise QS-1 card
                                                                ;SIO/DARTs, baud rates etc

0117'                           FOREVER::       ;repeat next sequence forever

                                        ;input from RS-232 Port 0
0117'  CD 0070'                 STAT0:: CALL    INSA0           ;check input status 0
011A'  CA 0117'                         JP      Z,STAT0         ;wait for data available
011D'  CD 0097'                         CALL    INCA0           ;yes, get data
0120'  32 014A'                         LD      (CHAR),A        ;save it in scratchpad

                                        ;output to RS-232 Port 1
0123'  CD 00C4'                 STAT1:: CALL    OUTSB0          ;check output status 1
0126'  C2 0123'                         JP      NZ,STAT1        ;wait for TBMT, etc (busy)
0129'  3A 014A'                         LD      A,(CHAR)        ;retrieve character
012C'  CD 00EE'                         CALL    OUTCB0          ;output it

                                        ;input from RS-232 Port 2
012F'  CD 007E'                 STAT2:: CALL    INSA1           ;check input status 2
0132'  CA 012F'                         JP      Z,STAT2         ;wait for data available
0135'  CD 00A5'                         CALL    INCA1           ;yes, get data
0138'  32 014A'                         LD      (CHAR),A        ;save it

                                        ;output to RS-232 Port 3
013B'  CD 00D2'                 STAT3:: CALL    OUTSB1          ;check output status 3
013E'  C2 013B'                         JP      NZ,STAT3        ;wait for TBMT etc (busy)
0141'  3A 014A'                         LD      A,(CHAR)        ;retrieve character
0144'  CD 00FE'                         CALL    OUTCB1          ;output it
0147'  C3 0117'                         JP      FOREVER         ;loop forever

                                ;---------------------------------------------------------------
                                SUBTTL SCRATCHPAD, STACK
                                PAGE
```

```
                        ;*****************************************************************************
                        ;SUBTTL SCRATCHPAD, STACK ALLOCATION
                        ;*****************************************************************************


                        ;change the ORG statement below to suit a suitable address
                        ;for the stack and scratchpad in your program - must be RAM
                        ;of course

                        ORG     $

014A'                   SCRATCH::

                        ;RAM for scratchpad
014A'                   CHAR::  DS      1       ;temporary storage for a character

                        ;RAM for stack
014B'                           DS      20      ;stack (more than enough for this
                                                ;this demonstration, but take more
                                                ;if you need it)
015F'                   STACK::         ;top of stack

                        ;-----------------------------------------------------------


                                END
```

SCRATCHPAD, STACK

Macros:

Symbols:

BAUD0	0004	BAUD1	0005	BD0050	0000	BD0075	0001
BD0110	0002	BD0134	0003	BD0150	0004	BD0300	0005
BD0600	0006	BD1200	0007	BD1800	0008	BD19K2	000F
BD2000	0009	BD2400	000A	BD3600	000B	BD4800	000C
BD7200	000D	BD9600	000E	BDRATE	0000	BRA0	0005
BRA1	000E	BRAB0	00E5	BRAB1	007E	BRB0	000E
BRB1	0007	CARDNO	0000	CHAR	014AI'	CTRLA	0001
CTRLB	0003	CTS0A	0003'	CTS0B	000E'	CTS1A	001B'
CTS1B	0026'	CTSEND	0033'	DATAA	0000	DATAB	0002
DEMO	0111I'	FOREVE	0117I'	INCA0	0097I'	INCA1	00A5I'
INCB0	009EI'	INCB1	00ACI'	INCH	00B3I'	INITQS	0033I'
INSA0	0070I'	INSA1	007EI'	INSB0	0077I'	INSB1	0085I'
INSTAT	008CI'	OSTAT	00D9I'	OUTCA0	00E6I'	OUTCA1	00F6I'
OUTCB0	00EEI'	OUTCB1	00FEI'	OUTCH	0106I'	OUTSA0	00BDI'
OUTSA1	00CBI'	OUTSB0	00C4I'	OUTSB1	00D2I'	QSCTRL	0007
QSPORT	0000	SCRATC	014AI'	SIO0	0000	SIO1	0001
STACK	015FI'	STAT0	0117I'	STAT1	0123I'	STAT2	012FI'
STAT3	013BI'						

No Fatal error(s)

A>